AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of distributing workload in a workflow management system comprising the steps of:

during a calibration mode, executing plural instantiations of a test process to identify load index parameters;

calculating a load index based on the load index parameters for each of a plurality of engines ~~engine~~ of the workflow management system, wherein each load index reflects a workload of its associated engine, wherein the load index corresponds to an average activity execution delay; and

distributing workload across the plurality of engines in response to the load indices in a load sensitive mode.

2. (Previously Presented) The method of claim 1:

wherein identifying the load index parameters comprises identifying a single engine nominal activity execution delay (C) when no concurrent activities are executing and an activity execution latency factor (λ), wherein λ is a function of a number of concurrently executing activities.

3. (Previously Presented) The method of claim 2 wherein calculating the load index comprises calculating the load index for each engine *j* as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing processes at the time activity *i* is executing, wherein λ_i, is an execution latency rate for activity *i*.

4. (Previously Presented) The method of claim 2 wherein calculating the load index comprises calculating the load index for each engine j as a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i.

5. (Previously Presented) The method of claim 1 wherein distributing the workload comprises re-directing incoming process requests to another engine.

6. (Previously Presented) The method of claim 1 wherein distributing the workload comprises re-distributing queued processes to another engine.

7. (Previously Presented) The method of claim 1 wherein distributing the workload comprises prioritizing a source engine for distributing workload from based on a maximum differential workload.

8. (Currently Amended) The method of claim 1 wherein distributing the workload comprises identifying a target engine to which workload is to be distributed ~~for distributing workload to~~ based on a maximum differential workload.

9. (Original) A method of distributing workload in a workflow management system comprising the steps of:

a) calculating a load index for each engine of the workflow management system, wherein each load index reflects a workload of its associated engine;

b) operating in a load insensitive workload distribution mode for distributing processes until a maximum differential load index exceeds a pre-determined threshold; and

c) operating in a load sensitive workload distribution mode for distributing processes until all processes have completed execution once the maximum differential load index exceeds the pre-determined threshold.

10. (Original) The method of claim 9 wherein processes are round-robin distributed in the load insensitive workload distribution mode.

11. (Original) The method of claim 9 wherein step a) further comprises the step of calculating the load index for each engine *j* as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing processes at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*, wherein C is a single engine nominal activity execution delay when no concurrent activities are executing.

12. (Original) The method of claim 9 wherein step a) further comprises the step of calculating the load index for each engine *j* as a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing activities at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*.

13. (Original) The method of claim 9 wherein step c) further comprises the step of re-directing incoming process requests to another engine.

14. (Original) The method of claim 9 wherein step c) further comprises the step of re-distributing queued processes to another engine.

15. (Original) The method of claim 9 wherein step c) further comprises the step of prioritizing a source engine for distributing workload from based on a maximum differential workload.

16. (Original) The method of claim 9 wherein step c) further comprises the step of identifying a target engine for distributing workload to based on a maximum differential workload.

17. (Currently Amended) A method of distributing workload in a workflow management system comprising the steps of:

a) switching from a load insensitive workload distribution mode to a load sensitive workload distribution mode for distributing processes when a maximum differential load index exceeds a first pre-determined threshold, T1; and

b) switching from the load sensitive workload distribution mode to the load insensitive workload distribution mode for distributing processes when the maximum differential load index is less than a second pre-determined threshold, T2.

18. (Previously Presented) The method of claim 17 wherein T1=T2.

19. (Previously Presented) The method of claim 17 wherein T1>T2.

20. (Original) The method of claim 17 wherein step a) further comprises the step of calculating a load index for each engine j as a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing processes at the time activity i is executing, wherein λ_i is an execution latency rate for activity i, wherein C is a single engine activity nominal execution delay when no concurrent activities are executing.

21. (Original) The method of claim 17 wherein step a) further comprises the step of calculating a load index for each engine j as a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein k is a total number of activities completed within a pre-determined time period for engine j, wherein N_i is the number of other concurrently executing activities at the time activity i is executing, wherein λ_i is an execution latency rate for activity i.

22. (Previously Presented) The method of claim 1, further comprising providing a definition of activities in the test process such that for each activity, a resource execution time is much less than an engine execution time, the resource execution time representing an execution time of a resource to perform work represented by the respective activity, and the engine execution time representing an execution time of the respective engine in performing the activity.

23. (Previously Presented) A workflow management system, comprising:

plural workflow engines;

workload monitors to compute load indices for the workflow engines, wherein each load index reflects a workload of the corresponding workflow engine; and

a load balancer to:

operate in a load insensitive workload distribution mode for distributing processes among the workflow engines in a first distribution fashion that is insensitive to the load indices until at least one difference between load indices of the workflow engines exceeds a first threshold; and

after the at least one difference between load indices exceeds the first threshold, operate in a load sensitive workload distribution mode for distributing processes among the workflow engines in a second distribution fashion that is sensitive to the load indices until at least one of:

(1) all processes have completed execution; and

(2) the at least one difference between load indices of the workflow engines is less than a second threshold.

24. (Previously Presented) The workflow management system of claim 23, wherein the load index for each engine *j* is a total average activity execution delay $L(j) = C + \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing processes at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*, wherein C is a single engine activity nominal execution delay when no concurrent activities are executing.

25. (Previously Presented) The workflow management system of claim 23, wherein the load index for each engine *j* is a relative average activity execution delay $L(j) = \frac{1}{k}\sum_{i=1}^{k} N_i \lambda_i$, wherein *k* is a total number of activities completed within a pre-determined time period for engine *j*, wherein N_i is the number of other concurrently executing activities at the time activity *i* is executing, wherein λ_i is an execution latency rate for activity *i*.

26. (New) The method of claim 1, wherein the plural instantiations of the test process are executed during the calibration mode to increase loading on each workflow engine to enable identification of the load index parameters.